

DATA MANAGEMENT FRAMEWORK FOR POLICY MANAGEMENT

5

CROSS-REFERENCE TO RELATED APPLICATION(S)

10

This application claims the benefit of U.S. Provisional Application No. 60/246,264 filed on November 6, 2000, the content of which is incorporated herein by reference.

FIELD OF THE INVENTION

15

This application relates generally to network management frameworks, and more particularly to a data management framework that provides a meta-model for expressing a network and relationships between entities in the model in a dynamic manner.

BACKGROUND OF THE INVENTION

20

Policy management is the application of abstract policies to control the services provided by a network switch. Policy is the association of conditions and actions, based on the attributes of the switch, that enforce certain behaviors in the switch, resulting in a specific service.

25

30

35

Because of the complexity and ranges of values for switching attributes to enforce specific services, the switching attributes need to be organized in a form that can be effectively supported on the network elements involved in the delivery of the services. Although network management system (NMS) solutions exist in the prior art which could be used for policy management, these solutions are undesirable because they generally map their concept of a network into statistically constructed virtual information stores, referred to as managed information bases (MIB). This mapping is often accomplished either through translators that translate an input of some

modeling language such as, for example, ASN.1 (Abstract Syntax
5 Notation One), or by hand coding all statistically typed model
elements, data, and behavior.

Existing network management architectures therefore do not
employ dynamic loading of element definitions and runtime
10 interrogation of types, nor do they generally use descriptive
meta-data information to elicit specific behaviors and
configuration of network modeling based on typing information.
Furthermore, current architectures usually bind network
management to typing the entire framework with respect to
15 protocol, allowing the framework and model to represent, for
example, ATM but not IP, carrier voice but not data, and IP/ATM
enterprise but not carrier ATM/frame relay.

Accordingly, there is a need for a network management
framework that provides a meta-model for expressing a network
20 and relationships between entities in the model in a dynamic
matter. The network management framework should allow data
representations and behavior, topology relationships, and
application services including policy management services, to
be dynamically assembled into a manageable model of a network
25 without restrictions as to typing, size, growth, and/or
performance.

SUMMARY OF THE INVENTION

30 In one embodiment, the present invention is directed to a
data management framework for a data communications network
including a client and a server in communication with the
client. The server provides a dynamically constructed model of
elements of the data communications network. The model provides
35 a uniform programming interface for allowing the client to

5 dynamically access the elements and dynamically add new elements in performing network management functions.

10 In another embodiment, the present invention is directed to a data management framework for a data communications network where the data management framework includes an application service providing application functions for the network, a modeling tool for creating a dynamic model of elements associated with the application service, and a uniform programming interface providing dynamic access to the elements for performing network management functions.

15 In a further embodiment, the present invention is directed to a data management framework for policy management where the data management framework includes a dynamically constructed model of a plurality of policy-related elements. The model provides a uniform programming interface for allowing dynamic access of the policy-related elements and dynamic addition of
20 new policy-related elements in performing network management functions.

25 In another embodiment, the present invention is directed to a method for data management in a data communications network. The method includes the steps of providing an application service for the network, creating a dynamic model of elements associated with the application service, and providing a uniform programming interface for providing dynamic access to the elements for performing network management
30 functions.

BRIEF DESCRIPTION OF THE DRAWINGS

35 These and other features, aspects and advantages of the present invention will be more fully understood when considered with respect to the following detailed description, appended claims, and accompanying drawings where:

10003904-10001

5 FIG. 1 is a schematic block diagram of various layers of a data management framework used for dynamically creating a network information model according to one embodiment of the invention;

10 FIG. 2 is a block diagram of various components and modules of the data management framework of FIG. 1 according to one embodiment of the invention; and

FIG. 3 is an exemplary object oriented diagram of a model context according to one embodiment of the invention.

15 DETAILED DESCRIPTION OF THE SPECIFIC EMBODIMENTS

FIG. 1 is a schematic block diagram of various layers of a data management framework used for dynamically creating a network information model according to one embodiment of the invention. Although the layers of the framework illustrated in
20 FIG. 1 are described in terms of various software layers and/or components, a person skilled in the art should recognize that one or more layers of the framework may also be implemented in middleware.

25 The various layers of the data management framework preferably include a sessions layer 10, application services layer 20, information models layer 30, support services layer 40, and a platform layer 50. The sessions layer 10 preferably provides connection for software systems and/or human users
30 (collectively referred to as clients) for accessing the application services, information models, and/or support services provided by respectively the application services layer 20, information models layer 30, and/or support services layer 40. Preferably, a connection is established via public
35 application programming interfaces (APIs) implemented via CORBA

5 (Common Object Request Broker Architecture), RMI (Remote Method Indication), or other remote interfaces and transport protocols that are conventional in the art. Once a connection is established, a session is preferably created. Sessions preferably control security and user access, log transactions, maintain connection attributes, assist in fault recovery, and
10 the like.

The application services layer 20 includes services that provide application feature functionality, such as, for example, policy management. According to one embodiment of the
15 invention, application services may be dynamically added and associated with features of the network information model as well as to other services provided by the framework.

The information models layer 30 includes information models used to create the data models associated with the application
20 services. Such information models may include conventional models such as, for example, Common Information Model (CIM), Managed Information Base (MIB), or the like. The information models are preferably containers of object instances that collectively model the information and behavior of real entities
25 in a real world management system, such as, for example, a policy management system.

The support services layer 40 provides support technologies for implementing software functionality and operating system
30 services for the data management framework. Support services include protocol stacks, databases, world wide web, graphics, transactions, fault detection/handling, resource management, and/or the like.

The platform layer 50 preferably includes a software
35 operating system and associated hardware for executing the

5 application services in the application services layer 20, including policy management. Preferably, the software operating system is implemented in the JAVA programming language.

10 FIG. 2 is a block diagram of the various components and modules of the data management framework according to one embodiment of the invention. The data management framework is preferably provided by a server 102 in communication with a client 100. The client 100 preferably includes connections from external software to the server 102 via the sessions layer 10. Such connections include graphic user interfaces (GUIs) 104, web connections 106, other application/management systems 108, and the like.

15 The server 102 preferably provides the sessions layer 10, application services layer 20, information models layer 30, and support services layer 40 of FIG. 1. The server 102 may reside in one processor/computer system or in multiple processors/computer systems. Furthermore, within the context of a single computer, the server 102 may include one or more Java Virtual Machines.

20 The server 102 preferably provides various controllers 110, 112, 114 within the session layer 10 where each controller controls a connection to the client 100. The controllers may control user access, user profile, transaction auditing, and the like.

25 The server 102 further provides one or more application services in the application services layer 20 such as, for example, a policy service 115 and other services 117 conventional in a data communication network. The elements of each application service are preferably represented as object

30
35

instances that are self-descriptive. The object instances are
5 referred to as managed objects (MOs) 120, 121.

A software bus 132, 134 allows the interconnecting of
various MOs to each other and to the corresponding application
service. The software bus may be implemented, for example,
10 using the InfoBus technology developed by Lotus.

Each MO is preferably an object instance implemented using
object oriented programming techniques that includes a
definition (class), name, attributes, and relationships with one
or more other MOs. The relationship between one more other MOs
15 is referred to as an association. Associations may be
named/identified, and include conditions, constraints, and other
forms or attributes describing a relationship between the MOs.
In the example of policy management, the MOs preferably model
different policies, actions, conditions, network elements,
20 directory servers, and other policy driven equipment and
software in a network.

The MOs associated with a particular application service
belong to a data model 116, 118 constructed using the
information models in the information models layer 30. Each
25 data model 116, 118 provides context, or containing environment,
in which the MOs reside and interact within software systems and
external systems. The data models are also referred to as a
model contexts.

Each model context has an application specific
functionality and service. Each model context allows control
and interaction over the objects inside the container. Any new
model context that is introduced or started may discover other
model contexts and enlarge the features of the network. Model
35 contexts advertise existence of the contents, publish attributes

and behavior, and control access. In this regard, each model
5 context provides a uniform API to the client 100 for interacting
with the model context in a dynamic manner. The network to
framework path may therefore be described in a dynamic fashion,
without static language restrictions or stoppage of processing.

Preferably, each model context exports behavior as a
10 combination of the context's public API, the context's name
service, the API of the application services and any attributes
or behaviors of the MOs publicly exported. A context has the
ability to advertise its attributes, actions and services as
15 well as those of its children. Once advertised, other
components may make use of the advertised information.

Each model context 116, 118 that makes up the network
information model may run stand-alone or in collaboration with
other model contexts. Thus, the entire network information
20 model may be made up of individual models that are distributed,
providing scalability and availability to the network management
solution.

The application services encompassed by the model contexts
25 116, 118 preferably utilize the support services in the support
services layer 40 in performing their functions. The support
services preferably implement the functionality shared by one
or more application service. For policy management, the support
services may include lightweight directory access protocol
30 (LDAP) databases 122, SNMP 124, database management services
126, error handling 128, and other services 130 such as, for
example, auto-discovery, and the like.

FIG. 3 is an exemplary object oriented diagram of a model
context according to one embodiment of the invention. According
35 to one embodiment of the invention, the model context and its

associated components and classes are implemented based on the
5 BeanContext API created by Sun Microsystems, Inc. BeanContext
is a pattern which provides a uniform API to access and use a
hierarchy of container classes and their children.

A particular model context preferably includes a base class
10 referred to as a model context class 140. The model context
class preferably provides default attributes, behavior, and
associations for a model context. Model contexts include one
or more application services 142, such as, for example, a policy
service 144 having a public API 150 and a policy databus 152.
15 A policy model context 146 is derived from the model context
class 140. The policy context 146 is associated with the
context's public API 148.

Model contexts contain information and behavior represented
by MOs. MOs preferably include a base class referred to as a
20 managed object entity (MOE) class 154 and a managed object
implementation (MOI) class 156. The MOE class 154 preferably
provides default attributes, behavior, and associations for an
MO. The MOI class 156 preferably describes how an MO is
implemented. For example, the MOI class 156 defines the
25 interface and attributes used to describe the services offered
by the MO, or the characteristics of the element represented by
the MO. A policy MO 158 is preferably derived from the MOI
class.

30 An MO may be created and reside in only one model context.
The MO may have associations with other MOs residing in the same
or different model context. An MO is preferably made public
outside the context via a naming service or an application
service using the application service's API.

35

5 An association class 160 preferably expresses a relationship between two or more MOs. The association class 160 preferably includes attributes that describe the qualification, constraints, rules of an association between two or more MOs, allowing other components and tools to operate based on this information.

10 Although this invention has been described in certain specific embodiments, those skilled in the art will have no difficulty devising variations which in no way depart from the scope and spirit of the present invention. Thus, the present
15 embodiments of the invention should be considered in all respects as illustrative and not restrictive, the scope of the invention to be indicated by the appended claims and their equivalents rather than the foregoing description.

20

25

30

35